****Java Developer Internship 2025

Project Name: Expense Tracker And Binary To Decimal Converter

Name: Pawar Rajani Baliram

Intern ID: VN-JD-4W293

Email: [rajanipawar6229@gmail.com](mailto:rajanipawar6229@gmail.com)

Week 2 Task:

Goal: Design a Java program to convert temperatures between Celsius, Fahrenheit, and Kelvin scales.

Structure:

-User Interface: Design a console-based or GUI interface using Java frameworks (e.g., Swing, JavaFX, or a simple CLI) tailored to the application's functionality and user experience.

-Core Logic Implementation: Implement the core logic and functionality of the application using clean, modular, and reusable Java code.

-Data Management: If applicable, design a system for storing and managing data (e.g., user inputs, application states, scores, or records). Use in-memory data structures or databases as per requirements.

-Error Handling and Validation: Include proper input validation, error handling, and exception management to ensure robust application behavior.

-Reports/Feedback (Optional): Add functionality to generate and display summaries, reports, or feedback based on user interaction or processed data.

Here's the documentation for your Temperature Converter application:

**Project Name: Temperature Converter**

**Name:** Pawar Rajani Baliram  
**Intern ID:** VN-JD-4W293  
**Email:** [rajanipawar6229@gmail.com](mailto:rajanipawar6229@gmail.com)

**1. Project Overview:**

The **Temperature Converter** is a simple console-based Java application that allows users to convert temperatures between different units. The application supports the following conversions:

* Celsius to Fahrenheit
* Fahrenheit to Celsius
* Celsius to Kelvin
* Kelvin to Celsius
* Fahrenheit to Kelvin
* Kelvin to Fahrenheit

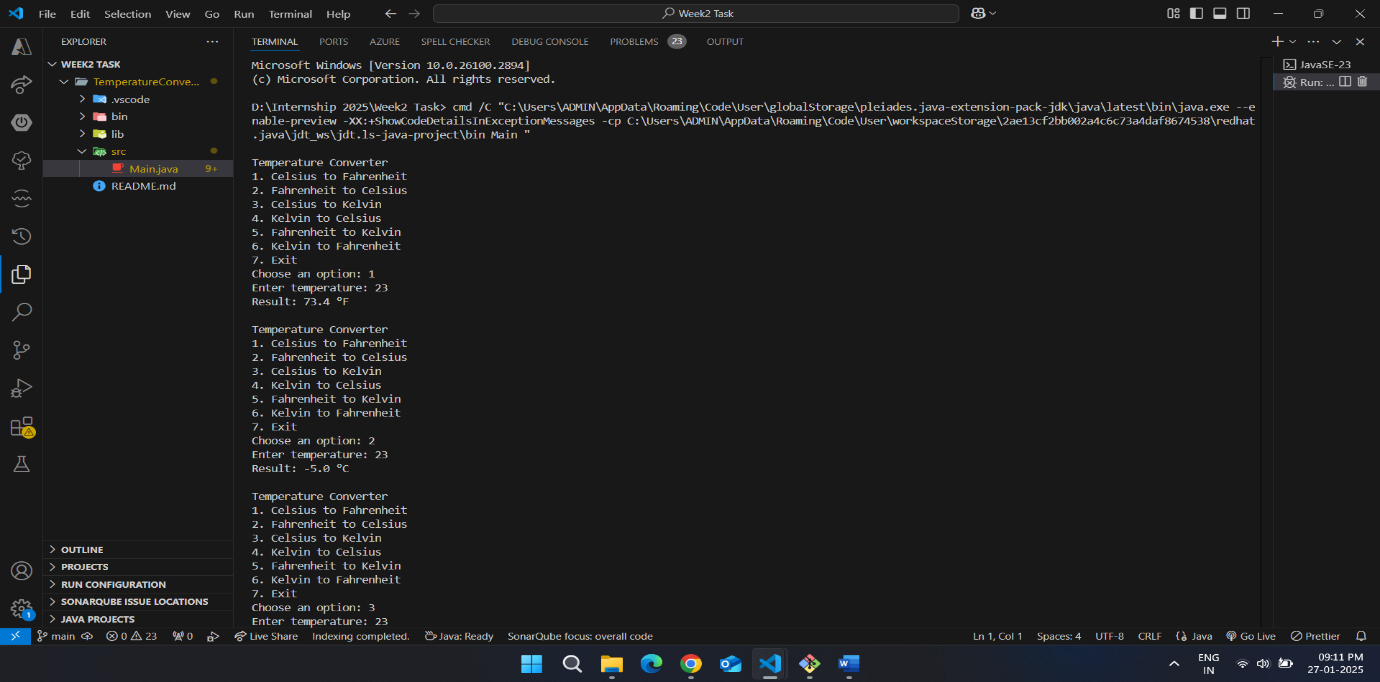
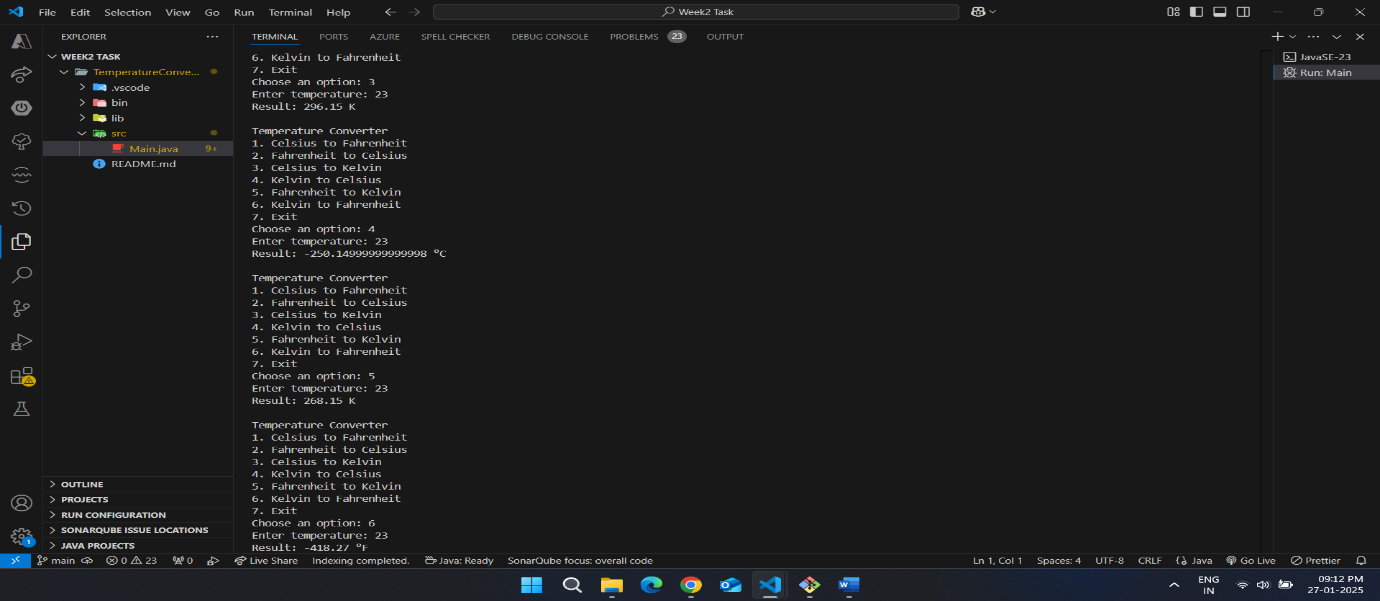
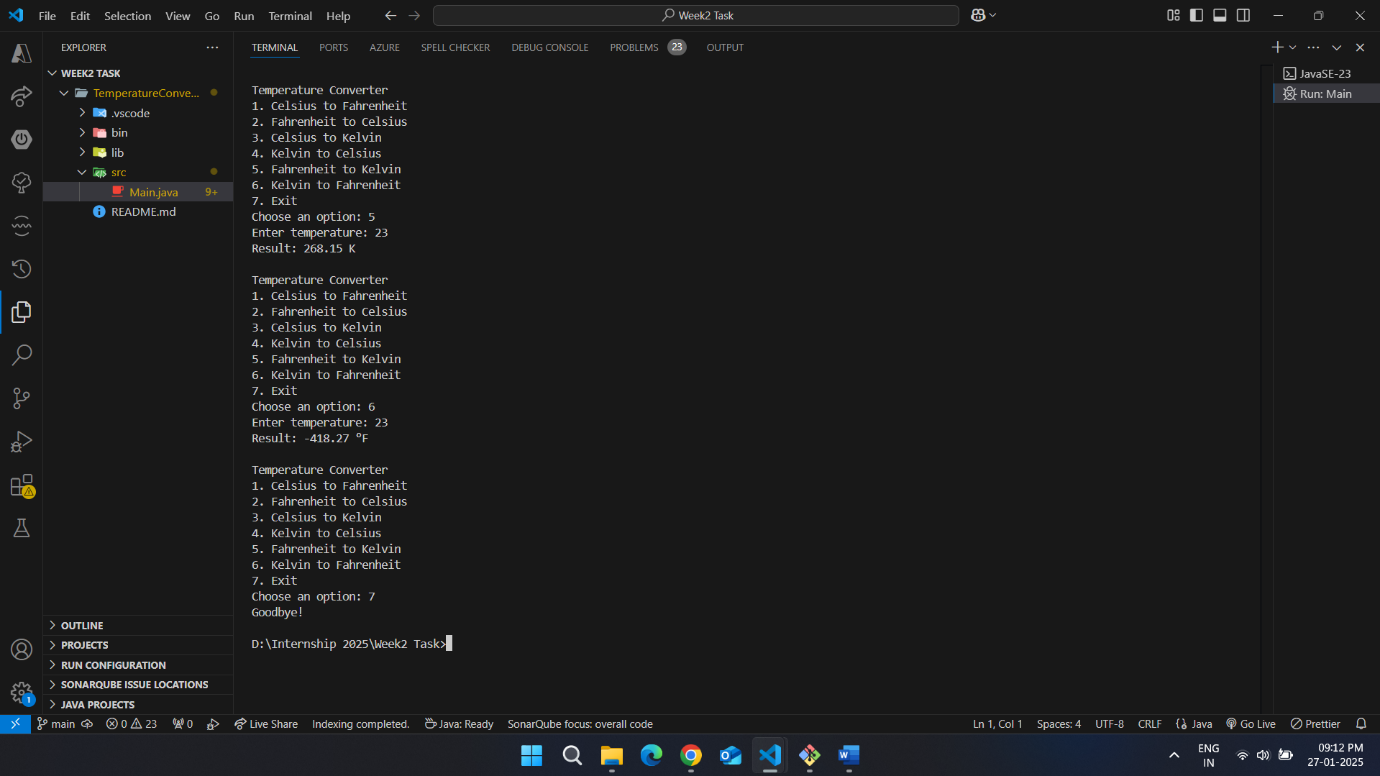
It includes input validation, error handling, and exception management to ensure robust behavior during operation.

**2. Features:**

* **Temperature Conversion:**
  + Convert temperatures between Celsius, Fahrenheit, and Kelvin.
* **Input Validation:**
  + Ensures the user provides a valid menu choice (1-7) and a valid temperature value.
* **Error Handling:**
  + Catches and handles exceptions caused by invalid input and prompts the user to re-enter correct values.
* **Exit Option:**
  + The application allows the user to exit the program gracefully by selecting option 7.

**3. Code Structure:**

* **Main Class:**
  + **main:** The main method that runs the application in a loop. It displays the menu, handles user input, and performs the necessary conversion.
* **Helper Methods:**
  + **getValidChoice:** Ensures the user enters a valid menu choice between 1 and 7.
  + **getValidTemperature:** Ensures the user enters a valid temperature value. It catches invalid inputs and prompts for re-entry.

**4.Output:**   

**8. Error Handling:**

* **Invalid Option Selection:** If the user enters a number outside the valid range (1-7), the program will prompt them to choose a valid option.
* **Invalid Temperature Input:** If the user enters an invalid temperature (non-numeric value), the program will display an error message and prompt the user to input a valid number.